Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MeowID: A Dual-Expert Retrieval System for Individual Cat Identification

Zhangchi Hu

HuZhangchi@mail.ustc.edu.cn

University of Science and Technology of China https://orcid.org/0009-0008-4030-825X

Yi Shang

Hefei University of Technology

Haocheng Yang

Northwestern Polytechnical University

Qiwei Hu

Beijing Forestry University

Yuzheng Li

Sun Yat-sen University

Research Article

Keywords: Animal Re-Identification, Image Retrieval, Deep Metric Learning

Posted Date: September 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10881972/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

MeowID

简体中文

MeowID: A Dual-Expert Retrieval System for Individual Cat Identification

Zhangchi Hu[*,†1,2], **Yi Shang**[*2], **Haocheng Yang**[*4,2], **Qiwei Hu**[*5], **Yuzheng Li**[*3]

[1]Department of Electronic Engineering and Information Science, University of Science and Technology of China
[2]School of Intelligent Software Engineering, Hefei University of Technology
[3]School of Software Engineering, Sun Yat-sen University
[4]School of Computer Science, Northwestern Polytechnical University
[5]College of Biological Sciences and Technology, Beijing Forestry University

[*]**Equal Contribution** [†]**Project Leader**



Abstract

We present **MeowID**, a dual-expert retrieval system for individual cat identification from photographs. Existing metric-learning and animal re-identification methods enable embedding-based gallery matching, but their performance is often affected by variations in pose, viewpoint, and facial visibility in unconstrained images. MeowID integrates complementary whole-cat and facial recognition experts within a face-priority retrieval framework. When a usable aligned face is available, the face representation is augmented with a validation-guided whole-cat hint that provides complementary global appearance information; otherwise, the system falls back to whole-cat retrieval. This design exploits fine-grained facial cues while preserving robust identity recognition when facial evidence is unavailable. Route-specific embedding galleries further allow new individuals to be enrolled without retraining the recognition models. We also introduce **Individual Cats in the Wild (ICW)**, an open-source annotated dataset and benchmark for individual cat retrieval. Experiments against representative classification, metric-learning, and animal re-identification baselines show that MeowID achieves more accurate and robust cat retrieval.

Keywords: Animal Re-Identification, Image Retrieval, Deep Metric Learning

Resources: Project Page Code Dataset

Contents

1 Introduction . . . 3

2 Related Work . . . 3
- 2.1 Individual Animal Re-Identification . . . 3
- 2.2 Face-Based Animal Identification . . . 3
- 2.3 Multi-Cue Representation Learning and Retrieval . . . 4

3 Method . . . 4
- 3.1 System Overview . . . 4
- 3.2 Face Localization, Alignment, and Routing . . . 4
- 3.3 DINOv3 Dual Recognition Experts . . . 5
- 3.4 Validation-Guided Shared Whole-Cat Hint . . . 5
- 3.5 Joint Training and Loss Definition . . . 6
- 3.6 Dynamic Teacher Update and Consistent Model Selection . . . 6
- 3.7 Deployment and Retrieval . . . 7

4 Dataset . . . 8
- 4.1 Data sources . . . 8
- 4.2 Automatic collection and filtering . . . 9
- 4.3 Duplicate removal and manual curation . . . 9
- 4.4 Dataset split and Face-available subset . . . 9

5 Experiments . . . 9
- 5.1 Datasets . . . 9
- 5.2 Experimental setup . . . 10
- 5.3 Main results . . . 11
- 5.4 Comparison with the Legacy Method . . . 12
- 5.5 Visualization Analysis . . . 13
- 5.6 Efficiency analysis . . . 13

6 Conclusion . . . 14

A Complete Implementation Details . . . 17
- A.1 Hyperparameter Configuration . . . 17
- A.2 Face Alignment . . . 17

B ICW Dataset Construction . . . 17
- B.1 Dataset Collection Prompt . . . 17
- B.2 Filtering and Manual Annotation Guidelines . . . 18

C Cat-Face Detector . . . 18
- C.1 Training Data and Annotation . . . 18
- C.2 Model and Training Details . . . 19
- C.3 Detection Performance . . . 19

D Additional Visualization Analysis . . . 19

1 Introduction

Individual animal identification plays an important role in applications such as pet management, lost animal recovery, and digital animal archives. Unlike conventional image classification, where the goal is to recognize predefined categories, individual identification requires distinguishing between visually similar individuals and supporting scalable registration of new identities. With the development of deep metric learning and retrieval-based recognition systems, embedding-based approaches have become an effective solution for large-scale identification, where images are encoded into feature representations and matched against a gallery through similarity search.

Cats present unique challenges for individual identification in real-world scenarios. Although individual cats often exhibit distinctive visual characteristics, their appearance can vary significantly across photographs due to changes in pose, viewpoint, illumination, occlusion, and image quality. Therefore, relying on a single visual representation may limit recognition performance in practical application environments.

Identity-related information may appear at different visual scales: whole-cat appearance provides cues such as coat patterns, torso markings, and tail structure across diverse viewpoints, whereas facial regions often contain more localized and fine-grained identity information but may not always be visible or reliably captured. In this work, we present **MeowID**, a dual-expert retrieval system for individual cat identification. MeowID combines separately parameterized whole-cat and face experts in a face-priority pipeline: facial evidence is used when a reliable aligned face is available, while whole-cat recognition provides complementary appearance information and fallback coverage for unconstrained images. The system uses route-specific embedding galleries, allowing new identities to be enrolled without retraining the recognition models.

We further introduce **Individual Cats in the Wild (ICW)**, an annotated benchmark for individual cat retrieval with identity-level annotations and an identity-disjoint evaluation protocol. Together, MeowID and ICW provide a practical framework and benchmark for scalable individual cat identification.

2 Related Work

2.1 Individual Animal Re-Identification

Individual animal re-identification aims to distinguish and retrieve the same individual across images based on individual-level phenotypic characteristics. Earlier approaches often relied on species-specific markings or local visual descriptors, while recent methods increasingly learn discriminative image embeddings using deep representation and metric learning. WildlifeDatasets [1] systematized evaluation across a broad collection of animal re-identification datasets and introduced MegaDescriptor, a large-scale representation model specialized for individual animals. Jiao et al. [8] further considered generalization to unseen animal categories with UniReID, while subsequent multi-species models such as MiewID [13] and recent animal ReID frameworks [7] explored transferable representations across diverse species and datasets. Despite this progress, most existing approaches represent an observation primarily through a unified image-level embedding. For cats in unconstrained photographs, however, identity cues may appear at different spatial scales and can vary substantially with pose and viewpoint. Our work therefore considers whole-cat and facial appearance as complementary sources of identity information.

2.2 Face-Based Animal Identification

Facial appearance provides localized and often fine-grained cues for individual animal recognition. Early studies demonstrated the feasibility of adapting deep face recognition techniques to individual dogs [12], while subsequent work has extended facial recognition to different animal species. More recently, PetFace [17] introduced a large-scale animal face identification benchmark covering diverse families and evaluated both face re-identification and verification. Reliable face-based recognition also depends on facial localization and geometric normalization; animal-specific landmark datasets and detectors, such as CatFLW [11], have facilitated face alignment under variations in appearance and pose. Nevertheless, a face-centric system cannot provide complete coverage in unconstrained imagery, where the face may be occluded, poorly localized, or

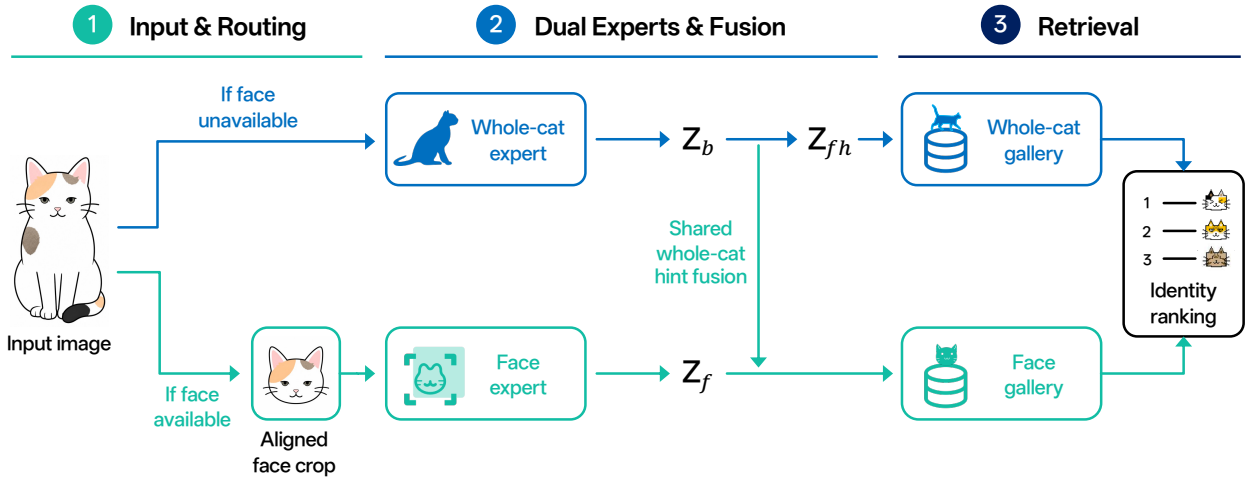


Figure 1 Model pipeline of MeowID.

outside the field of view. This limitation motivates recognition systems that can exploit fine-grained facial evidence when it is reliable while retaining non-facial identity cues when it is unavailable.

2.3 Multi-Cue Representation Learning and Retrieval

Combining global and localized visual evidence has been extensively studied in image retrieval and re-identification. In person ReID, part-based approaches such as PCB [19] complement holistic appearance with localized representations, while transformer-based models such as TransReID [6] further improve robustness to viewpoint and spatial variation through patch-level representation learning. Related ideas have recently appeared in animal identification: WildFusion [2], for example, combines global deep-descriptor similarity with local feature-matching scores through calibrated score fusion. These studies demonstrate the benefit of combining complementary identity cues, but they typically fuse multiple representations or similarity signals within a common retrieval procedure. In contrast, MeowID adopts an asymmetric, route-aware design in which facial and whole-cat evidence play distinct roles rather than being treated as interchangeable representations in a single retrieval space.

3 Method

3.1 System Overview

As shown in Fig. 1, MeowID adopts a dual-expert architecture with two separately parameterized embedding encoders: a face expert and a whole-cat expert. The face expert captures localized, fine-grained identity information, whereas the whole-cat expert captures complementary cues such as coat patterns, torso markings, and tail structure.

MeowID follows a face-priority routing strategy. When a usable aligned face is available, the face representation serves as the primary identity embedding and is augmented by a shared whole-cat hint. When facial evidence is unavailable, the system falls back to whole-cat retrieval. This design preserves the discriminative advantage of facial appearance while maintaining recognition coverage in unconstrained images.

3.2 Face Localization, Alignment, and Routing

Given an input image I^b, the face localization module detects the cat face and predicts 9 facial landmarks. The landmark set contains the left and right eye centers, the mouth center, and 3 landmarks on each ear. When a valid face is detected, we construct an aligned face crop I^f using a PetFace-style [17] similarity alignment based on the eye centers and mouth center. The transformation normalizes translation, in-plane rotation, and scale while preserving facial proportions.

We consider a face crop usable when the image can be decoded and a valid alignment can be constructed from the predicted landmarks. We first estimate the similarity transformation from the eye centers and mouth center. When these landmarks do not support a stable transformation, we fall back to a landmark-based square crop using the available facial landmarks. If neither procedure produces a valid crop, the face is treated as unavailable. Exact alignment templates and fallback parameters are centralized in the implementation configuration rather than embedded in the routing logic.

Face availability directly determines which recognition route is used:

$$\text{route}(I) = \begin{cases} \text{face expert with the shared whole-cat hint}, & \text{if } I^f \text{ is usable}, \\ \text{whole-cat expert}, & \text{otherwise}. \end{cases} \tag{1}$$

Images without a usable face remain valid identity-labeled samples for the whole-cat branch, but do not participate in face-expert training or face-only evaluation.

3.3 DINOv3 Dual Recognition Experts

Both experts use a DINOv3 ViT backbone pretrained on LVD-1689M [18]. The two experts share the architecture and initialization family but have separate trainable parameters.

For each expert, the class token and a generalized-mean-pooled summary of the patch tokens are combined by a learned two-way gate. We denote the GeM pooling exponent by p_{GeM}. Let $\mathbf{h}_{\text{cls}}$ and $\mathbf{h}_{\text{patch}}$ denote the two summaries. The pooled feature is

$$\boldsymbol{\omega} = \text{softmax}\left(P\left([\mathbf{h}_{\text{cls}}, \mathbf{h}_{\text{patch}}]\right)\right), \qquad \mathbf{h} = \omega_1 \mathbf{h}_{\text{cls}} + \omega_2 \mathbf{h}_{\text{patch}}, \tag{2}$$

where P is a small multilayer perceptron. A linear projection followed by batch normalization maps $\mathbf{h}$ to a cat embedding.

Let $f_b(\cdot;\theta_b)$ and $f_f(\cdot;\theta_f)$ denote the whole-cat and face encoders. Their normalized embeddings are

$$\mathbf{z}_b = \frac{f_b(I^b;\theta_b)}{\|f_b(I^b;\theta_b)\|_2}, \qquad \mathbf{z}_f = \frac{f_f(I^f;\theta_f)}{\|f_f(I^f;\theta_f)\|_2}. \tag{3}$$

3.4 Validation-Guided Shared Whole-Cat Hint

The live whole-cat parameters θ_b continue to change during training. Using the live embedding as the face hint would make the hint distribution drift with every optimization step and would also leave the independently selected face checkpoint coupled to an arbitrary whole-cat snapshot. We instead maintain training-only teacher parameters $\widetilde{\theta}_b$, defined as a frozen copy of the whole-cat encoder that has achieved the best validation mAP so far. For a sample with a usable face, the hint embedding is

$$\widetilde{\mathbf{z}}_b = \text{sg}\left(\frac{f_b(I^b;\widetilde{\theta}_b)}{\left\|f_b(I^b;\widetilde{\theta}_b)\right\|_2}\right), \tag{4}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. The teacher is kept in evaluation mode, receives no gradients, and is excluded from the optimizer.

A learnable linear projection maps the hint toward the face feature space:

$$\mathbf{p}_b = \frac{W_p \widetilde{\mathbf{z}}_b}{\|W_p \widetilde{\mathbf{z}}_b\|_2}. \tag{5}$$

The hint contribution is controlled by a sample-dependent scalar gate,

$$g = \sigma\left(G\left([\mathbf{z}_f, \mathbf{p}_b, |\mathbf{z}_f - \mathbf{p}_b|, \mathbf{z}_f \odot \mathbf{p}_b]\right)\right), \tag{6}$$

where G is a multilayer perceptron, σ is the sigmoid function, $[\cdot]$ denotes concatenation, and $\odot$ denotes element-wise multiplication. A second multilayer perceptron A converts the unprojected teacher embedding into a residual. The final face-hint embedding is

$$\mathbf{z}_{fh} = \frac{\mathbf{z}_f + \alpha_{\text{hint}}\, gA(\widetilde{\mathbf{z}}_b)}{\|\mathbf{z}_f + \alpha_{\text{hint}}\, gA(\widetilde{\mathbf{z}}_b)\|_2}, \tag{7}$$

where α_{hint} is the residual scale. This formulation preserves the face embedding as the primary representation while allowing whole-image context to provide a gated correction.

3.5 Joint Training and Loss Definition

The two experts are trained in the same optimization run with separate classification heads. We use Sub-center ArcFace with K sub-centers per identity [3, 4]. For route $r \in \{b, f\}$, let $\mathbf{w}^r_{c,k}$ denote the kth normalized sub-center of identity c. The similarity to identity c is defined by the maximum cosine similarity over its sub-centers, $\cos\theta^r_{i,c} = \max_k (\mathbf{z}^r_i)^\top \mathbf{w}^r_{c,k}$, where $\mathbf{z}^b_i = \mathbf{z}_{b,i}$ and $\mathbf{z}^f_i = \mathbf{z}_{fh,i}$. Given identity label y_i, the branch-specific identity loss is

$$\mathcal{L}^r_{\text{id}} = -\frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \log \frac{\exp\big(s_{\text{arc}} \cos(\theta^r_{i,y_i} + m_{\text{arc}})\big)}{\exp\big(s_{\text{arc}} \cos(\theta^r_{i,y_i} + m_{\text{arc}})\big) + \sum_{c \neq y_i} \exp\big(s_{\text{arc}} \cos\theta^r_{i,c}\big)}. \tag{8}$$

Here, s_{arc} and m_{arc} denote the ArcFace logit scale and additive angular margin, respectively. $\mathcal{S}_b = \mathcal{B}$ contains the complete mini-batch, whereas $\mathcal{S}_f = \mathcal{B}_f \subseteq \mathcal{B}$ contains only samples with usable faces.

We additionally use the batch-hard triplet loss [16]. For each valid anchor, the hardest positive is the most distant sample of the same identity in the current mini-batch, while the hardest negative is the closest sample of a different identity. Let these distances be $d^{r,+}_i$ and $d^{r,-}_i$, respectively, and let $\mathcal{V}_r$ denote the set of anchors for which both are available. The route-specific triplet loss is

$$\mathcal{L}^r_{\text{tri}} = \frac{1}{|\mathcal{V}_r|} \sum_{i \in \mathcal{V}_r} \left[d^{r,+}_i - d^{r,-}_i + \mu_{\text{tri}} \right]_+, \tag{9}$$

with $\mathcal{L}^r_{\text{tri}} = 0$ when $\mathcal{V}_r$ is empty.

Two additional regularizers constrain the whole-cat hint on the face route. The alignment term encourages the projected teacher feature $\mathbf{p}_{b,i}$ to remain compatible with the raw face embedding, while the preservation term discourages the fused representation from deviating excessively from the face representation:

$$\begin{aligned} \mathcal{L}_{\text{align}} &= \frac{1}{|\mathcal{B}_f|} \sum_{i \in \mathcal{B}_f} \left[1 - \cos(\mathbf{p}_{b,i}, \text{sg}(\mathbf{z}_{f,i}))\right], \\ \mathcal{L}_{\text{preserve}} &= \frac{1}{|\mathcal{B}_f|} \sum_{i \in \mathcal{B}_f} \left[1 - \cos(\mathbf{z}_{fh,i}, \text{sg}(\mathbf{z}_{f,i}))\right]. \end{aligned} \tag{10}$$

The complete training objective is

$$\begin{aligned} \mathcal{L} = {} & \lambda^b_{\text{id}} \mathcal{L}^b_{\text{id}} + \lambda^b_{\text{tri}} \mathcal{L}^b_{\text{tri}} \\ & + \mathbb{I}[|\mathcal{B}_f| > 0] \Big(\lambda^f_{\text{id}} \mathcal{L}^f_{\text{id}} + \lambda^f_{\text{tri}} \mathcal{L}^f_{\text{tri}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\text{preserve}} \mathcal{L}_{\text{preserve}} \Big). \end{aligned} \tag{11}$$

The whole-cat terms are averaged over samples in $\mathcal{B}$, whereas the face-route terms are computed only over $\mathcal{B}_f$.

3.6 Dynamic Teacher Update and Consistent Model Selection

Whole-cat and face validation are performed sequentially after each epoch. Let $M_b(e)$ be whole-cat validation mAP at epoch e, computed using every validation image and the whole-cat gallery. The selected whole-cat epoch after epoch e is

$$e^*_b(e) = \arg\max_{1 \leq j \leq e} M_b(j). \tag{12}$$

Whenever $M_b(e)$ exceeds the previous best, we atomically update

$$\widetilde{\theta}_b \leftarrow \theta_b^{(e)}, \qquad e_b^* \leftarrow e. \tag{13}$$

Face validation for the same epoch is then run using the refreshed teacher. It includes only validation samples with usable aligned faces and compares their face-hint embeddings with the face-hint gallery.

A face checkpoint selected under an older teacher is not a valid deployment candidate after Eq. (13). Therefore, whenever the best-body teacher changes, we reset the face-checkpoint selection window. Let $\mathcal{E}(e_b^*)$ denote the validation epochs evaluated with the final teacher $\widetilde{\theta}_b = \theta_b^{(e_b^*)}$. The selected face epoch is

$$e_f^* = \arg\max_{e \in \mathcal{E}(e_b^*)} \mathrm{mAP}_{fh}^{\mathrm{val}}\left(e; \widetilde{\theta}_b = \theta_b^{(e_b^*)}\right). \tag{14}$$

This rule allows e_b^* and e_f^* to differ while guaranteeing that the selected face expert and fusion modules were validated against the same whole-cat parameters selected for fallback inference. The checkpoint metadata records the teacher epoch, and export verifies exact equality between the stored teacher and the selected whole-cat encoder before deployment.

3.7 Deployment and Retrieval

After training, the deployed parameters are assembled as

$$\theta_b^{\mathrm{deploy}} = \theta_b^{(e_b^*)}, \qquad (\theta_f^{\mathrm{deploy}}, \theta_{\mathrm{fusion}}^{\mathrm{deploy}}) = (\theta_f^{(e_f^*)}, \theta_{\mathrm{fusion}}^{(e_f^*)}). \tag{15}$$

The training-only teacher $\widetilde{\theta}_b$ is then removed. Because it is verified to equal $\theta_b^{(e_b^*)}$, the single deployed whole-cat encoder $f_b(\cdot; \theta_b^{\mathrm{deploy}})$ reproduces the hint used to select the face checkpoint. In a single forward pass, its output can be reused as the fallback embedding and, when a face is available, as the input to the face fusion module.

The system maintains route-specific enrollment galleries. Every registered image contributes a whole-cat embedding $\mathbf{z}_b$ to the whole-cat gallery. If the image also has a usable aligned face, it contributes a fused embedding $\mathbf{z}_{fh}$ to the face gallery. Registering a new identity requires only embedding extraction and gallery insertion; the recognition models are not retrained.

At query time, Eq. (1) is applied. A query with a usable face is represented by $\mathbf{z}_{fh}$ and compared only with the face-hint gallery. A query without a usable face is represented by $\mathbf{z}_b$ and compared only with the whole-cat gallery. All embeddings are L_2-normalized, so image-level similarity is the inner product,

$$s(\mathbf{z}_q, \mathbf{z}_g) = \mathbf{z}_q^\top \mathbf{z}_g, \tag{16}$$

where $\mathbf{z}_q$ and $\mathbf{z}_g$ always come from the same route.

When identity c has multiple enrollment images, their scores are combined by log-mean-exp aggregation,

$$S_c(\mathbf{z}_q) = \log\left(\frac{1}{|\mathcal{G}_c^r|} \sum_{\mathbf{z}_g \in \mathcal{G}_c^r} \exp(s(\mathbf{z}_q, \mathbf{z}_g))\right), \tag{17}$$

where r is the selected route and $\mathcal{G}_c^r$ is the corresponding gallery for identity c. Registered identities are ranked by S_c.

The resulting design preserves full-image coverage while exploiting the stronger facial representation whenever it is available. More importantly, validation-guided teacher updates and compatibility-constrained face selection ensure that training-time hinting and deployment-time inference use the same whole-cat parameters without maintaining a second whole-cat model copy.

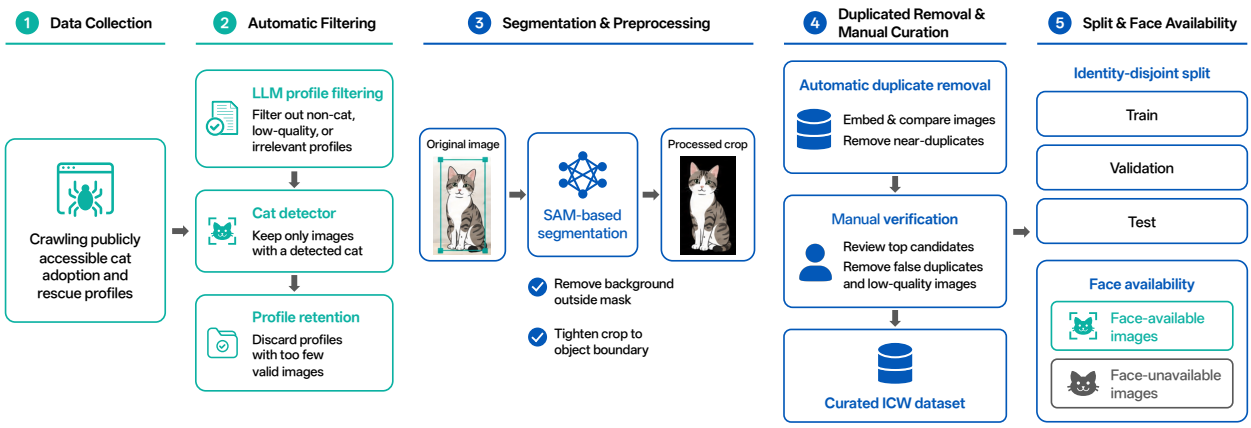


Figure 2 Data pipeline of the Individual Cats in the Wild(ICW) Dataset.

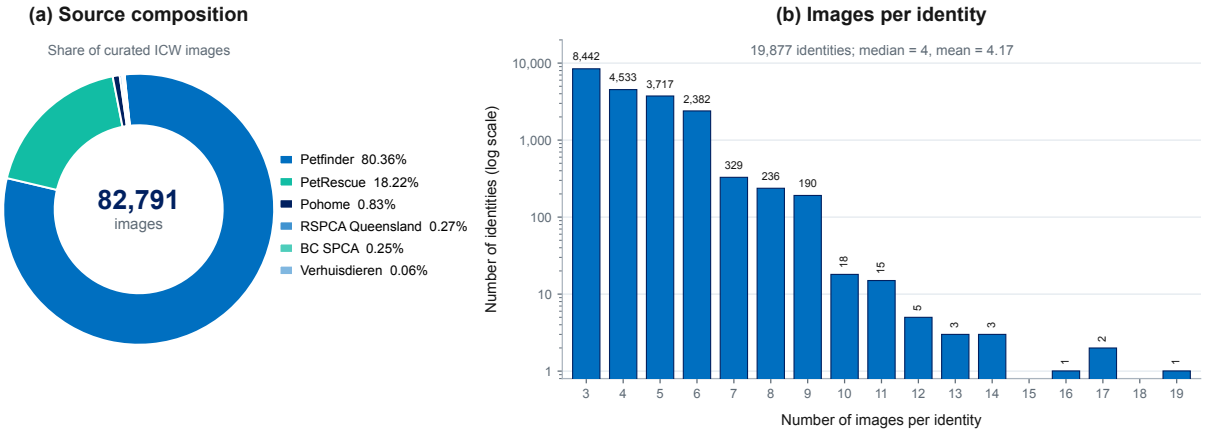


Figure 3 Composition and sample distribution of the curated ICW dataset after automatic filtering and manual quality control. (a) Distribution of the 82,791 images across the six source platforms. (b) Number of identities for each number of images per identity.

4 Dataset

As illustrated in fig. 2, we construct the Individual Cats in the Wild(ICW) dataset for individual cat recognition from publicly accessible cat adoption and rescue profiles. Unlike datasets in which identity labels are inferred from image similarity or cat names, ICW uses the unique profile identifier provided by each source platform as the initial identity label. All images collected from the same profile are therefore treated as different observations of the same individual cat. This strategy provides multiple images per identity while avoiding ambiguity caused by cats sharing the same name.

4.1 Data sources

The dataset is collected from 6 cat adoption and animal-rescue platforms, including Petfinder, PetRescue, Pohome, RSPCA Queensland, BC SPCA, and Verhuisdieren. These sources contain images captured under diverse conditions, including different viewpoints, poses, illumination, backgrounds, cameras, and levels of occlusion. The source composition and the number of images per identity in the final curated dataset are shown in Figure 3. Petfinder and PetRescue constitute the majority of the dataset, while the remaining sources increase its geographical and visual diversity. Most identities contain between 3 and 6 images, reflecting the low-shot nature of practical cat-identification scenarios.

4.2 Automatic collection and filtering

We apply a two-stage automatic filtering procedure to reduce identity noise. First, the profile description is analyzed by a LLM to reject listings that explicitly refer to multiple cats, bonded pairs, or a mixture of different animals. Second, a cat detector is applied independently to each image. An image is retained only if exactly one cat is detected. Profiles containing fewer than 3 valid images after this filtering step are discarded.

To reduce contamination from the surrounding scene, the detected bounding box is used as a prompt for SAM-based [9] foreground segmentation. Pixels outside the predicted cat mask are replaced with black, and the result is tightly cropped to the mask boundary. Consequently, the ICW images preserve the appearance of the complete visible cat while suppressing potentially misleading background information.

4.3 Duplicate removal and manual curation

Duplicate removal is performed at both the automatic and manual stages. Each profile is represented by a fingerprint consisting of its metadata and image hashes. File-level MD5 hashes identify exact duplicates, whereas 64-bit difference hashes are used to identify perceptually similar images.

After automatic collection and preprocessing, all retained data undergo a second manual annotation and verification pass. During this stage, annotators re-check the identity assignments and remove remaining duplicate or near-duplicate images. Visually low-quality or otherwise unusable samples are also excluded. This manual pass is important because automatic detection and hash-based matching cannot reliably identify all identity conflicts, repeated web images, or poor-quality samples.

The resulting dataset contains 82,791 images from 19,877 individual cats. Each identity contains between 3 and 19 images, with a median of 4 images and an average of 4.17 images per identity. This long-tailed, low-shot setting closely reflects practical cat-identification scenarios, where only a small number of reference images are usually available for each animal.

4.4 Dataset split and Face-available subset

As shown in tab. 1, ICW is divided into training, validation, and test sets at the identity level. The identity sets are mutually disjoint. This prevents images of the same cat from appearing in multiple splits and ensures that evaluation measures generalization to previously unseen identities.

Table 1 Statistics of the ICW splits and face availability.

Split	IDs	Images	Face	No face
Train	18,877	77,094	73,924	3,170
Validation	500	2,851	2,720	131
Test	500	2,846	2,717	129
Total	19,877	82,791	79,361	3,430

To support the face-expert branch described in Section 3, A trained face localization module is applied to every curated ICW image. An image is marked as face-available whenever ECPose produces a valid cat-face detection and keypoint prediction; no additional confidence-based sample filtering is applied. Overall, 79,361 images, corresponding to 95.86% of ICW, contain a detectable cat face. The remaining 3,430 images are retained for the whole-cat expert but are not used to train or evaluate the face expert.

The face-available training subset contains 73,924 images from 18,873 identities. Four training identities contain no detectable face and are therefore excluded only from face-expert training. In the test set, 2,717 images are routed to the face expert, whereas the remaining 129 images are processed exclusively by the whole-cat expert.

5 Experiments

5.1 Datasets

ICW. We use the identity-disjoint ICW splits described in Section 4. The test set contains 2,846 images from 500 unseen identities, of which 2,717 have usable aligned faces and 129 are processed through the whole-cat fallback route. Face crops are generated using the alignment procedure described in Section 3.2.

Table 2 Retrieval performance on the ICW full test set and the face-available subset.

	Full Set				Face Subset			
Model	Top-1	Top-5%	Top-50%	mAP	Top-1	Top-5%	Top-50%	mAP
MegaDescriptor [1]	56.85	89.78	**99.75**	65.45	58.81	91.20	**99.82**	67.37
NekoNet [21]	57.24	89.88	94.45	67.48	59.96	94.15	98.93	70.68
PetFace [17]	71.86	90.62	94.31	78.08	75.27	**94.92**	98.79	81.79
MeowID-Base	**75.93**	**90.72**	96.52	**80.45**	**78.80**	93.19	98.23	**83.32**

Table 3 Animal verification results on PetFace [17], measured by AUC (%). Models are trained per animal family unless marked by † and evaluated on the official test split.

Method	Cat	Chimp	Chinchilla	Degus	Dog	Ferret	Guinea pig	Hamster	Hedgehog	Parakeet	Sparrow	Pig	Rabbit	Avg.
Trained on Other Dataset														
ImageNet-R50 [15][5]	81.58	73.67	70.80	70.69	97.18	67.95	85.19	74.13	75.01	90.45	79.49	82.42	86.23	79.60
CLIP [14]	85.10	70.70	73.42	74.66	91.86	70.89	80.67	73.91	77.65	83.19	76.52	80.02	83.55	78.63
MegaDescriptor [1]	87.30	83.01	77.85	77.79	93.75	76.53	88.36	77.91	78.74	89.76	78.64	88.08	90.38	83.70
Trained on PetFace														
PetFace-Softmax [17]	97.97	85.22	84.44	86.19	98.98	83.15	95.30	87.73	87.27	98.39	84.67	87.09	98.53	90.38
PetFace-Center [17]	50.20	81.60	80.96	81.34	59.64	78.77	92.10	83.59	85.26	96.91	80.54	82.58	97.01	80.81
PetFace-Triplet [17]	96.94	77.10	76.12	72.80	97.97	79.48	83.37	80.08	81.47	91.56	77.86	75.72	94.74	83.48
PetFace-ArcFace [17]	97.71	83.76	87.70	87.61	99.45	86.24	96.03	90.32	86.13	98.63	84.45	89.88	99.01	91.30
PetFace-ArcFace† [17]	98.04	83.20	89.63	86.79	99.01	88.44	94.14	92.50	87.86	98.40	89.38	92.38	98.45	92.17
MeowID-Base (Ours)	**99.04**	**88.74**	**94.35**	**91.66**	**99.61**	**93.85**	**96.91**	**94.78**	**91.34**	**99.22**	**89.93**	**95.67**	**99.10**	**94.94**

PetFace. PetFace [17] is a large-scale aligned animal-face dataset containing 1,012,934 images of 257,484 individuals from 13 animal families and 319 breeds. The released face crops have a resolution of 224×224. We use the official family-specific training lists, validation lists, and animal verification pairs. In contrast to the ICW retrieval protocol, PetFace verification evaluates whether two face images belong to the same previously unseen individual. We use this benchmark to test whether the embedding formulation learned by MeowID-Base generalizes beyond cats and across animal families with substantially different facial structures and dataset sizes.

5.2 Experimental setup

Implementation details. MeowID-Base is implemented in PyTorch and trained on 4 NVIDIA GeForce RTX 3090 GPUs. Both whole-cat and face experts adopt DINOv3 ViT-B/16 backbones [18] with 512-dimensional embeddings and 256×256 input resolution. Initialized from their respective single-branch ICW checkpoints, the two experts are jointly optimized for 30 epochs using AdamW with a base learning rate of 3×10^{-4}, weight decay of 10^{-2}, and a global batch size of 32. The learning rate is warmed up for 2 epochs and then cosine-decayed to 10^{-6}. Only the last six Transformer blocks are fine-tuned with branch-specific learning-rate multipliers. We employ Sub-center ArcFace (3 sub-centers, angular margin 0.35, scale 64) combined with the auxiliary and hint-compatibility losses introduced in Section 3. Validation runs after each epoch: the whole-cat expert checkpoint is selected by whole-cat validation mAP, while the face expert is chosen by face-subset validation mAP under the best validation whole-cat teacher.

For PetFace [17], we build on the MeowID-Base embedding encoder and train a separate DINOv3 ViT-B/16 ArcFace model for each animal family. Inputs are of 224×224 resolution, with a per-GPU batch size of 16 on four NVIDIA RTX 3090 GPUs. We use AdamW with an initial learning rate of 1.5×10^{-4}, weight decay of 0.05, one warm-up epoch, and cosine decay to 10^{-6}. Each model is trained for 10 epochs with per-epoch validation. For each family, the checkpoint is selected by AUC on the official validation split and evaluated on the official verification test pairs.

Retrieval protocol and metrics. We evaluate ICW as identity-level closed-set retrieval. Each validation or test image acts as a query, with all remaining images from the same split forming the gallery. As each identity may correspond to multiple gallery images, cosine similarities are aggregated into identity-level scores via log-sum-exp. For the zero-shot MegaDescriptor baseline, maximum aggregation is adopted based on its superior validation performance. We report Top-1 accuracy, Top-5% and Top-50% recall, and mean average

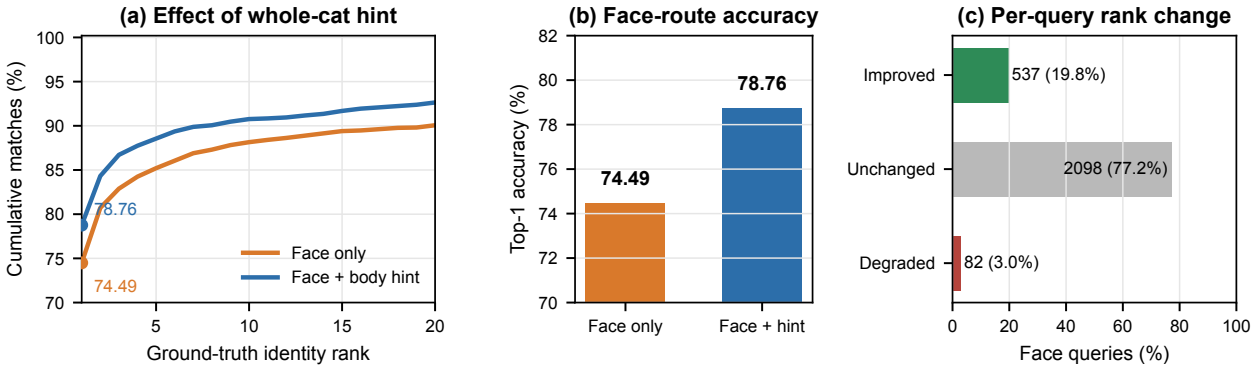


Figure 4 Quantitative visualization analysis on the ICW test set. (a) Cumulative matching curves of the face-only embedding and the face embedding augmented with the whole-cat hint. (b) Top-1 accuracy on face-available queries. (c) Per-query change in the ground-truth identity rank after introducing the whole-cat hint.

precision (mAP). A Top-p% hit is counted when the ground-truth identity ranks within the top p% of 500 candidates. The full-set protocol follows the hard router design: queries with valid faces are matched in the face embedding space, and the rest use the whole-cat embedding space. For PetFace verification, we follow the official protocol, computing cosine similarity for the provided positive and negative pairs; all verification identities are unseen during training. Performance is measured by AUC, with the reported average being the unweighted mean across 13 animal families.

Compared methods. We compare MeowID-Base with three representative embedding-based systems on ICW. MegaDescriptor [1] is evaluated in a zero-shot manner using its public Swin-L/384 checkpoint, with no ICW-specific weight updates. NekoNet [21] adopts an ImageNet-pretrained EfficientNetV2-B0 face encoder with 128-dimensional embeddings, trained on aligned ICW faces with its metric-learning objective over a four-stage 100-epoch schedule. PetFace [17] uses the authors' ResNet-50 ArcFace formulation, retrained on the same aligned ICW training faces. As both NekoNet and PetFace are face-only systems, queries without valid faces contribute zero to all retrieval metrics in the ICW full-set evaluation.

5.3 Main results

Individual-cat retrieval on ICW. Table 2 compares MeowID-Base with representative embedding-based systems on both the full ICW test set and its face-available subset. On the complete test set, MeowID-Base reaches 75.93% Top-1 accuracy and 80.45% mAP. It outperforms the strongest face-only baseline PetFace by 4.07 pp in Top-1 accuracy and 2.37 pp in mAP, with gains of 18.69 / 12.97 pp over NekoNet and 19.08 / 15.00 pp over zero-shot MegaDescriptor. This advantage is particularly meaningful for the full test set, as MeowID-Base produces valid predictions for all 129 queries without detectable faces, where face-only methods fail entirely.

Consistent trends appear on the face-available subset, where MeowID-Base achieves 78.80% Top-1 accuracy and 83.32% mAP, surpassing PetFace by 3.53 / 1.53 pp and NekoNet by 18.84 / 12.64 pp. While some baselines yield marginally higher Top-5% or Top-50% recall, MeowID-Base consistently leads in Top-1 accuracy and mAP, reflecting more accurate top-ranked results and better overall identity ranking. The consistent improvement on the face-only subset confirms that the gain does not stem solely from fallback coverage—the face branch itself learns more discriminative embeddings under the same aligned-face input protocol.

Animal verification on PetFace. Table 3 presents verification AUC results on PetFace. MeowID-Base attains an average AUC of 94.94% across 13 animal families, exceeding the independently trained PetFace-ArcFace baseline by 3.64 pp and the jointly trained version by 2.77 pp. It outperforms the independent baseline on every family, with notable gains for Ferret (+7.61), Chinchilla (+6.65), Pig (+5.79), Java sparrow (+5.48), and Hedgehog (+5.21), and also surpasses the joint model across all 13 families despite using separate



Figure 5 Representative hard-routing and retrieval cases on ICW. From top to bottom, the examples show a face-route prediction corrected by the whole-cat hint, a stable face match, a successful whole-cat fallback when no aligned face is available, and a failure shared by the face-only and hinted representations. Whole-cat images are resized with aspect-ratio preservation, with unused regions padded in black. Green and red borders denote correct and incorrect retrieved identities, respectively. GT rank denotes the rank of the ground-truth identity.

family-specific training. These results verify that the DINOv3 embedding scheme generalizes well to unseen identities, with advantages extending beyond the ICW cat retrieval task.

5.4 Comparison with the Legacy Method

Internal Cat Profile Dataset The Internal Cat Profile Dataset is constructed from two complementary sources. The first subset consists of identity-labeled cat images collected through a mini-program and provided by the "Duxing Maopu" team. The second subset contains additional identity-labeled cat images collected from publicly available Internet sources. This dataset is designed to evaluate performance under realistic operational scenarios. We merge the original training and validation partitions by identity and retain only identities containing at least six images. After filtering, the dataset contains 9,342 images from 417 individual cats. For each identity, we randomly select up to three images as queries while preserving at least five remaining images for training and registration. This results in 8,210 training/registration images and 1,132 query images.

To ensure a controlled comparison, MeowID uses the black-background cat images as registration samples. For the detection-based baseline, the same foreground cats are randomly composited with COCO 2017 images that contain no annotated cat instances. Both methods are evaluated using exactly the same identity split and query set. Retrieval is performed over 417 candidate identities; therefore, Top-5% and Top-50% correspond to Top-21 and Top-209, respectively.

Table 4 Comparison with Sysucats/Cat-face on the Internal Cat Profile Dataset.

Metric	Sysucats/Cat-face [20]	**MeowID**
Top-1 (%)	64.22	**90.81**
Top-5 (%)	94.17	**97.97**
Top-50 (%)	99.73	**99.91**
mAP (%)	73.03	**93.11**

Results We compare MeowID trained on ICW with the legacy detection-based identification pipeline from the Sysucats/Cat-face [20] project. The legacy pipeline first detects the cat using YOLO11m and then classifies the detected crop using a YOLO11m classification model. If no cat is detected, a separate whole-image classifier is used as the fallback. Both classification models are trained for 150 epochs, and the final checkpoints are evaluated without selecting models on the query set.

As shown in Table 4, MeowID substantially outperforms the legacy method, improving Top-1 accuracy by 26.59 pp and mAP by 20.09 pp. The improvement at Top-1 is considerably larger than that at broader retrieval ranges, indicating that MeowID is especially effective at placing the correct identity at the very top of the ranking.

5.5 Visualization Analysis

We further analyze MeowID-Base on ICW from three perspectives: the effect of the whole-cat hint, the behavior of the hard-routing mechanism, and the identity-level retrieval results. As shown in Figure 4, 2,717 of the 2,846 test images have an available face crop and are therefore processed by the face route. On these queries, incorporating the whole-cat hint increases the Top-1 accuracy of the face expert from 74.49% to 78.76%, corresponding to an absolute improvement of 4.27 pp. The cumulative matching curves show that the advantage is most pronounced at small retrieval depths, which are the most relevant operating points for practical identification.

The per-query analysis provides a more detailed view of this improvement. Introducing the whole-cat hint improves the ground-truth identity rank for 537 queries (19.8%), leaves it unchanged for 2,098 queries (77.2%), and degrades it for only 82 queries (3.0%). This asymmetric distribution indicates that the whole-cat representation usually preserves an already reliable face prediction, while correcting a substantial fraction of ambiguous face-only matches. Thus, the hint acts as complementary contextual information rather than replacing the more discriminative face representation.

Figure 5 illustrates representative outcomes of the routing mechanism. The whole-cat hint can correct an ambiguous face-only match while leaving an already confident face prediction unchanged. When no usable face is available, the router switches to the whole-cat expert and preserves retrieval coverage. The failure example shows that visually similar individuals remain challenging when neither facial nor global contextual cues are sufficiently discriminative.

Figure 6 focuses on the ranked identity lists produced by the two routes. Successful face-route examples place the correct identity at Rank-1, including a case corrected by the whole-cat hint, while a more ambiguous query retains the ground-truth identity within the Top-5. The whole-cat examples show analogous behavior when facial evidence is unavailable, ranging from a clear Rank-1 match to a more difficult lower-ranked retrieval.

5.6 Efficiency analysis

We evaluate the inference efficiency of the full MeowID system across four deployment backends: PyTorch, ONNX Runtime (CPU), ONNX Runtime (GPU), and TensorRT with FP16 acceleration. Experiments are conducted on the full 2,846-image ICW test set with a fixed batch size of 1, mimicking latency-sensitive online identification scenarios.

Table 5 End-to-end inference efficiency on ICW with batch size one. Latency is reported in ms/image.

Backend	Precision	Latency ↓	Throughput ↑
PyTorch GPU	FP32	94.23	10.61 img/s
ONNX Runtime CPU	FP32	478.67	2.09 img/s
ONNX Runtime GPU	FP32	79.88	12.51 img/s
TensorRT GPU	FP16	**60.00**	**16.66 img/s**

Measured latency covers the end-to-end pipeline, including cached image loading and decoding, preprocessing, cat-face keypoint detection, face alignment, whole-cat and face embedding extraction, and final hard routing. Model initialization and checkpoint loading are excluded. Each backend is warmed up with 20 images before measurement. GPU experiments use a single NVIDIA GeForce RTX 3090, while CPU experiments use one Intel Xeon Gold 6226R socket with 16 physical cores and 32 threads.

As shown in Table 5, ONNX Runtime GPU reduces the average end-to-end latency from 94.23 ms with PyTorch to 79.88 ms, corresponding to a 1.18× speedup. TensorRT achieves the lowest latency at 60.00 ms

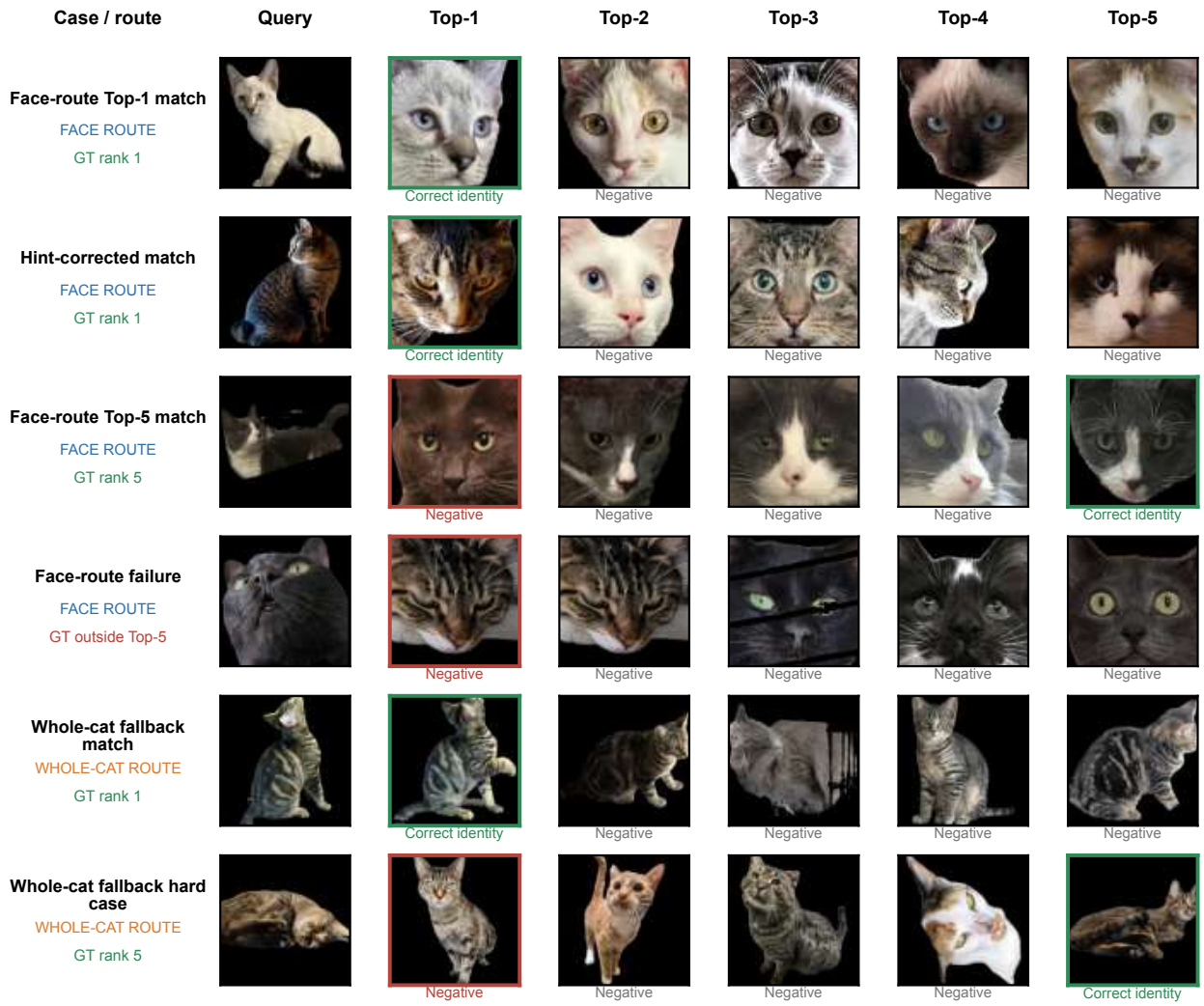


Figure 6 Top-5 identity retrieval examples on ICW. Each ranked identity is represented by its most similar gallery image. Green borders indicate the ground-truth identity, red borders indicate an incorrect Top-1 prediction, and black borders indicate the remaining retrieved candidates.

per image and the highest throughput at 16.66 images/s, yielding 1.57× and 1.33× speedups over PyTorch and ONNX Runtime GPU, respectively. This corresponds to latency reductions of 36.3% and 24.9%.

ONNX Runtime CPU is substantially slower at 478.67 ms per image, but enables GPU-free deployment for offline or resource-constrained settings. Overall, TensorRT provides the most efficient option for latency-sensitive deployment, while ONNX Runtime offers a portable alternative across CPU and GPU platforms.

6 Conclusion

We presented MeowID, a dual-expert retrieval system that combines face and whole-cat representations for individual cat identification. A face-priority hard router uses the face expert with a validation-guided whole-cat hint when a valid face is available and falls back to whole-cat retrieval otherwise. We also introduced ICW, an identity-disjoint benchmark with human annotation of high quality. MeowID-Base outperforms all existing methods on cat retrieval and verification tasks, which demonstrate the effectiveness and generality of its dual-expert design. The route-specific retrieval framework further enables robust identification under varying facial visibility while supporting efficient enrollment of new individuals.

Acknowledgement

As the project leader, I started this project in 2024. Its development was interrupted for a period due to various circumstances, and I am grateful that, after nearly two years, we have finally been able to bring it to completion. During this time, my first cat, "Lao Mao," went missing. Completing this project has therefore carried a personal meaning for me, and I hope this work can serve as a tribute to her.

I would also like to thank the "Duxing Maopu" project from Sun Yat-sen University for providing the initial data that supported this work. I am also sincerely grateful to my senior colleague Hebei Li and my PhD adviser Xiaoyan Sun for their support and help throughout my graduate studies.

References

[1] Vojtěch Čermák, Lukas Picek, Lukáš Adam, and Kostas Papafitsoros. Wildlifedatasets: An open-source toolkit for animal re-identification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 5953–5963, 2024. doi: 10.1109/WACV57701.2024.00585.

[2] Vojtěch Cermak, Lukas Picek, Lukáš Adam, Lukáš Neumann, and Jiří Matas. Wildfusion: Individual animal identification with calibrated similarity fusion. In Computer Vision – ECCV 2024 Workshops, volume 15624 of Lecture Notes in Computer Science, pages 18–36. Springer, 2025. doi: 10.1007/978-3-031-92387-6_2.

[3] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center ArcFace: Boosting face recognition by large-scale noisy web faces. In European Conference on Computer Vision (ECCV), pages 741–757. Springer, 2020. doi: 10.1007/978-3-030-58621-8_43.

[4] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10): 5962–5979, October 2022. ISSN 1939-3539. doi: 10.1109/tpami.2021.3087709. URL http://dx.doi.org/10.1109/TPAMI.2021.3087709.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[6] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 15013–15022, 2021.

[7] Saihui Hou, Panjian Huang, Zengbin Wang, Yuan Liu, Zeyu Li, Man Zhang, and Yongzhen Huang. Openanimals: Revisiting person re-identification for animals towards better generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 14369–14379, 2025.

[8] Bingliang Jiao, Lingqiao Liu, Liying Gao, Ruiqi Wu, Guosheng Lin, Peng Wang, and Yanning Zhang. Toward re-identifying any animal. In Advances in Neural Information Processing Systems, volume 36, 2023.

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

[10] Longfei Liu, Yongjie Hou, Yang Li, Qirui Wang, Youyang Sha, Yongjun Yu, Yinzhi Wang, Peizhe Ru, Xuanlong Yu, and Xi Shen. Edgecrafter: Compact vits for edge dense prediction via task-specialized distillation, 2026. URL https://arxiv.org/abs/2603.18739.

[11] George Martvel, Ilan Shimshoni, and Anna Zamansky. Automated detection of cat facial landmarks. International Journal of Computer Vision, 132(8):3103–3118, 2024. doi: 10.1007/s11263-024-02006-w.

[12] Thierry Pinheiro Moreira, Mauricio Lisboa Perez, Rafael de Oliveira Werneck, and Eduardo Valle. Where is my puppy? retrieving lost dogs by facial features. Multimedia Tools and Applications, 76(14):15325–15340, 2017. doi: 10.1007/s11042-016-3824-1.

[13] Lasha Otarashvili, Tamilselvan Subramanian, Jason Holmberg, J. J. Levenson, and Charles V. Stewart. Multispecies animal re-id using a large community-curated dataset. arXiv preprint arXiv:2412.05602, 2024.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. URL https://arxiv.org/abs/1409.0575.

[16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 815–823, 2015.

[17] Risa Shinoda and Kaede Shiohara. Petface: A large-scale dataset and benchmark for animal identification. In European Conference on Computer Vision (ECCV), volume 15076 of Lecture Notes in Computer Science, pages 19–36. Springer, 2025. doi: 10.1007/978-3-031-72649-1_2.

[18] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

[19] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In Proceedings of the European Conference on Computer Vision (ECCV), pages 480–496, 2018.

[20] SYSU Cats. cat-face: Cat face recognition for sysu cats. URL https://github.com/sysucats/cat-face. GitHub repository.

[21] tokudayo. Nekonet: Feature vector descriptor from cat face for verification and identification. URL https://github.com/tokudayo/NekoNet.

Appendix

A Complete Implementation Details

A.1 Hyperparameter Configuration

Table 6 reports the implementation settings that are not fully specified in the main text. Both recognition experts are initialized from their corresponding single-expert checkpoints and are then jointly optimized. Unless stated otherwise, the two branches use the same architectural settings.

A.2 Face Alignment

The cat-face localizer predicts nine landmarks in the following fixed order: left eye, right eye, mouth center, three landmarks on the left ear, and three landmarks on the right ear. Let the first three predicted points be q_l, q_r, and q_m. We estimate a least-squares similarity transform $T(x) = sRx + t$ that maps them to the normalized PetFace-style template

$$\bar{q}_l = (0.250000, 0.512291), \qquad \bar{q}_r = (0.750000, 0.511518), \qquad \bar{q}_m = (0.498841, 0.819070). \tag{18}$$

The transformed image is rendered at 256×256 using bicubic interpolation. Regions outside the source image are filled with RGB $(0, 0, 0)$ rather than reflected or replicated pixels.

The similarity transform is used whenever the three core landmarks are finite and the inter-eye distance is at least one pixel. If the transform is degenerate, we compute the axis-aligned bounding box of all finite landmarks, center a square on that box, set its side length to 1.25 times the larger landmark span, and resize it to 256×256. An image is marked face-available if either procedure produces a valid crop. For the offline ICW experiments, every row returned by the localizer with a valid detection is used: the detector score is stored as metadata but is not used to filter, weight, or rank samples.

B ICW Dataset Construction

B.1 Dataset Collection Prompt

The profile-level filter used the model identifier `doubao-seed-2-0-mini-260215`. The crawler inserted the complete profile metadata at `<profile-json>` and required a machine-parseable Boolean decision. The deployed prompt was written in Chinese; the following is a faithful English rendering that preserves its decision rules and output contract:

```
Please judge the animal-adoption information strictly according to the following requirements:

Decision criteria:

1. The animal must be a cat; the supplied image has already established that it is very likely
a cat.

2. There must be exactly one animal. If the description contains terms such as ‘‘Bonded pair’’,
‘‘Double’’, ‘‘Together’’, or ‘‘Pair’’, return false.

3. If the entry is mixed with any other animal, return false.

Input data: <profile-json>

Important restriction: output only a valid JSON string in one of the following forms:

{‘‘is_single_cat’’: true} or {‘‘is_single_cat’’: false}.

Do not include explanations, introductory text, or Markdown code-block markers.
```

The collection run supplied the structured profile metadata to this prompt; the optional image field supported by the service wrapper was not used in these crawler calls. Markdown fences were stripped before JSON parsing. A missing key, malformed response, or non-fatal API/parsing error was handled conservatively as `false`, so uncertain profiles did not enter the curated set.

B.2 Filtering and Manual Annotation Guidelines

Each source-platform profile identifier was used as the initial identity label. The automatic and manual filters were then applied as follows:

1. **Profile-level single-animal filtering.** The prompt above rejected descriptions of multiple cats, bonded pairs, or mixed-species listings.
2. **Image-level cat filtering.** A configured Ultralytics cat localizer was applied to every image. The released collection code resolves `YOLO26x-seg` as its first-choice model and uses a cat-confidence threshold of 0.25. An image was retained only when exactly one cat box remained.
3. **Foreground extraction.** The retained cat box was used as the prompt for SAM-based segmentation; the released configuration resolves `mobile_sam.pt` first. Pixels outside the selected mask were set to black, and the result was cropped to the tight mask boundary. Profiles with fewer than three valid segmented images were discarded.
4. **Automatic duplicate removal.** Exact image duplicates were identified with MD5. A 64-bit difference hash with a maximum Hamming distance of 6 was used for near-duplicate matching. In the primary Petfinder crawler, a listing was rejected when it shared at least three exact images with an existing listing. For listings from the same organization with the same normalized name and compatible breed and sex metadata, the additional rules were at least two exact images with description similarity ≥ 0.90, or at least three near-duplicate images with description similarity ≥ 0.97.
5. **Image-level manual review.** Annotators compared the original and masked images for each profile and removed repeated images, images containing the wrong individual or multiple animals, incorrect or severely incomplete masks, and images too blurred, occluded, or otherwise unusable for identification.
6. **Identity-level manual review.** After the image decisions, the annotator either approved or rejected the complete profile. The annotation interface prevented approval when fewer than three images remained. The final export retained only profiles marked `approved` and images not marked `remove`.

Four annotators completed the profile review. The annotation-stage export contained 19,887 approved profiles and 82,831 retained images; a final consistency and duplicate audit produced the reported ICW release of 19,877 identities and 82,791 images. Identities were then assigned to mutually disjoint training, validation, and test sets containing 18,877, 500, and 500 identities, respectively.

C Cat-Face Detector

C.1 Training Data and Annotation

The face localizer was trained from two compatible nine-landmark cat-face collections: a CSV-annotated image collection and a collection organized as `CAT_00`–`CAT_06` with per-image `.cat` landmark files. MD5 matching was used to transfer the landmark files to overlapping images and to avoid importing duplicate copies. The merged positive set used for the final mixed training run contained 19,994 images, randomly divided with seed 42 into 17,994 training images and 2,000 validation images.

Every positive image contains one annotated cat face. The nine points are the left eye, right eye, mouth center, three points along the left ear, and three points along the right ear. All source landmarks were encoded as visible ($v = 2$). The training box was derived from the minimum and maximum landmark coordinates and expanded by 20 pixels on each side, clipped to the image boundary. The data were converted to the COCO keypoint format.

To suppress false face detections on non-cat content, we added COCO 2017 images whose annotations contain no `cat` category. We sampled 5,000 images from `train2017` for training and 1,500 images from `val2017` for validation. These images have empty cat-face annotations and were deliberately retained by setting `filter_empty_gt=false`. The resulting mixed splits contain 22,994 training images and 3,500 validation images.

C.2 Model and Training Details

We use the ECPose-X configuration from EdgeCrafter[10], adapted from human pose estimation to a two-class, nine-keypoint cat-face task. The ViT-S+ adapter backbone has embedding dimension 384, six attention heads, projection dimension 256, and interaction indices $[10, 11]$. The hybrid encoder uses three 256-channel feature levels and a 256-dimensional hidden representation. The DETR-style pose decoder has four layers, eight heads, 60 queries, hidden dimension 256, feed-forward dimension 2048, and nine output keypoints. The complete model has 50.55M trainable parameters and operates on 640×640 inputs.

The matching costs for classification, keypoint regression, and OKS are 2, 10, and 4, respectively; the corresponding training-loss weights are also 2, 10, and 4. The focal-loss parameter is 0.25. The model is initialized from the preceding positive-only cat-face ECPose checkpoint and optimized for 100 epochs with AdamW. The main learning rate is 5×10^{-4}, the backbone learning rate is 2.5×10^{-6}, weight decay is 1.25×10^{-4}, and normalization and bias parameters receive zero weight decay. We use a 125-iteration linear warm-up, a MultiStep schedule with milestone 1250 and factor 0.5, gradient clipping at 0.1, synchronized batch normalization, EMA with decay 0.9999 and 2,000 warm-up updates, and automatic mixed precision. Training uses four RTX 3090 GPUs with batch size 8 per GPU (global batch size 32), seed 0, random horizontal flipping, and color jitter. The validation batch size is 8 globally. The final localizer is the validation-best stage-2 checkpoint.

C.3 Detection Performance

Table 7 reports COCO keypoint metrics on the 3,500-image mixed validation set. The best checkpoint was obtained after epoch 100 (zero-based log index 99). These values measure joint face localization and landmark quality; a separate bounding-box AP was not used for checkpoint selection.

When applied to ICW without additional score-based filtering, the localizer produced a valid detection and alignment for 79,361 of 82,791 images (95.86%). Split-wise face-availability statistics are reported in Table 1.

Table 7 Cat-face ECPose performance on the mixed validation set. AP and AR use the COCO OKS protocol with at most 20 detections per image.

Metric	Value (%)
AP @[.50 : .95]	96.78
AP @.50	99.10
AP @.75	96.50
AP (medium)	86.80
AP (large)	97.00
AR @[.50 : .95]	99.40
AR @.50	100.00
AR @.75	99.70

D Additional Visualization Analysis

Fig. 7, 8 show the additional qualitative results further demonstrate the robustness of MeowID (ICW-trained) under diverse appearance and pose variations on Internal Cat Profile Dataset. Across the presented cases, MeowID consistently retrieves the correct identity at Top-1, while competing methods occasionally rank the true match at lower positions. This advantage is particularly evident for challenging queries involving substantial changes in viewpoint, pose, scale, and visible body regions. These observations suggest that our method learns more identity-discriminative and appearance-invariant representations, leading to more reliable retrieval across diverse feline instances.

Table 6 Complete hyperparameter configuration of MeowID-Base. The learning-rate multipliers are relative to the base learning rate.

Component	Hyperparameter	Value
Input	Recognition input and normalization	256×256 RGB; ImageNet mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$
Input	Evaluation resize	Whole-cat: resize shorter side to 292 and center-crop to 256; face: resize and center-crop to 256
Recognition backbone	DINOv3 ViT-B/16	12 blocks, hidden dimension 768, 12 attention heads, patch size 16, four register tokens; last six blocks trainable
Token pooling	Global descriptor	Class token and trainable GeM patch pooling; $p_{\mathrm{GeM}} = 3$; two-way gated combination with hidden dimension 512
Embedding head	Projection	Linear layer followed by batch normalization; embedding dimension 512; dropout 0
Classification head	Sub-center ArcFace	Three sub-centers per identity; scale 64; angular margin 0.35; label smoothing 0
Shared hint	Projector and adapter	Linear $512 \to 512$ projector; adapter $512 \to 512 \to 512$ with GELU; projector initialized to identity and adapter output layer initialized to zero
Shared hint	Gate	Input $[z_f, p_b, \lvert z_f - p_b \rvert, z_f \odot p_b]$; MLP $2048 \to 512 \to 1$ with GELU and sigmoid; final-layer weights initialized to zero and bias to -2
Shared hint	Residual scale	$\alpha_{\mathrm{hint}} = 0.10$
Objective	Loss weights	$\lambda^b_{\mathrm{id}} = 1$, $\lambda^f_{\mathrm{id}} = 1$, $\lambda^b_{\mathrm{tri}} = 0.10$, $\lambda^f_{\mathrm{tri}} = 0.10$, $\lambda_{\mathrm{align}} = 0.20$, $\lambda_{\mathrm{preserve}} = 0.10$
Objective	Triplet loss	Batch-hard triplet loss with margin $\mu_{\mathrm{tri}} = 0.20$; anchors without both a positive and a negative in the mini-batch contribute zero
Optimizer	AdamW	Base learning rate 3×10^{-4}; weight decay 10^{-2}; $\beta = (0.9, 0.999)$
Optimizer	Learning-rate multipliers	Whole-cat expert 0.02; face expert 0.01; whole-cat and face classification heads 0.10; body projector, hint adapter, and hint gate 1.0
Schedule	Training schedule	30 epochs; two warm-up epochs starting at 0.1 of the target rate; cosine decay to 10^{-6}
Optimization	Numerical settings	Automatic mixed precision; gradient clipping at 5.0; one optimization step per mini-batch; no gradient accumulation
Sampling	Mini-batches	Ordinary shuffled sampling without a PK sampler; batch size 8 per GPU and global batch size 32 on four RTX 3090 GPUs
Whole-cat augmentation	Geometric	Random resized crop, scale $[0.70, 1.00]$ and aspect ratio $[0.75, 1.33]$; horizontal flip 0.30; affine probability 0.30, rotation $\pm 12^\circ$, translation 0.03, scale $[0.9, 1.1]$, shear $\pm 4^\circ$; perspective probability 0.15 and distortion 0.15
Whole-cat augmentation	Appearance	Color jitter $(0.25, 0.25, 0.15, 0.03)$; Gaussian blur probability 0.10; random erasing probability 0.15 with area $[0.02, 0.12]$ and aspect ratio $[0.3, 3.3]$
Face augmentation	Geometric and appearance	Random resized crop, scale $[0.90, 1.00]$ and aspect ratio $[0.95, 1.05]$; horizontal flip 0.50; affine probability 0.15, rotation $\pm 5^\circ$, translation 0.02, scale $[0.95, 1.05]$; color jitter $(0.20, 0.20, 0.15, 0.03)$; blur 0.10; random erasing 0.10
Validation	Frequency and selection	Validation after every epoch; log-mean-exp identity aggregation; mAP is maximized independently for the whole-cat and face routes, subject to the teacher-compatibility rule in Sec. 3.6
Retrieval	Ranking	Cosine similarity between L_2-normalized embeddings; log-mean-exp aggregation over enrollment images; self-image matches are masked
Reproducibility	Random seed	42 for MeowID-Base training and data loading

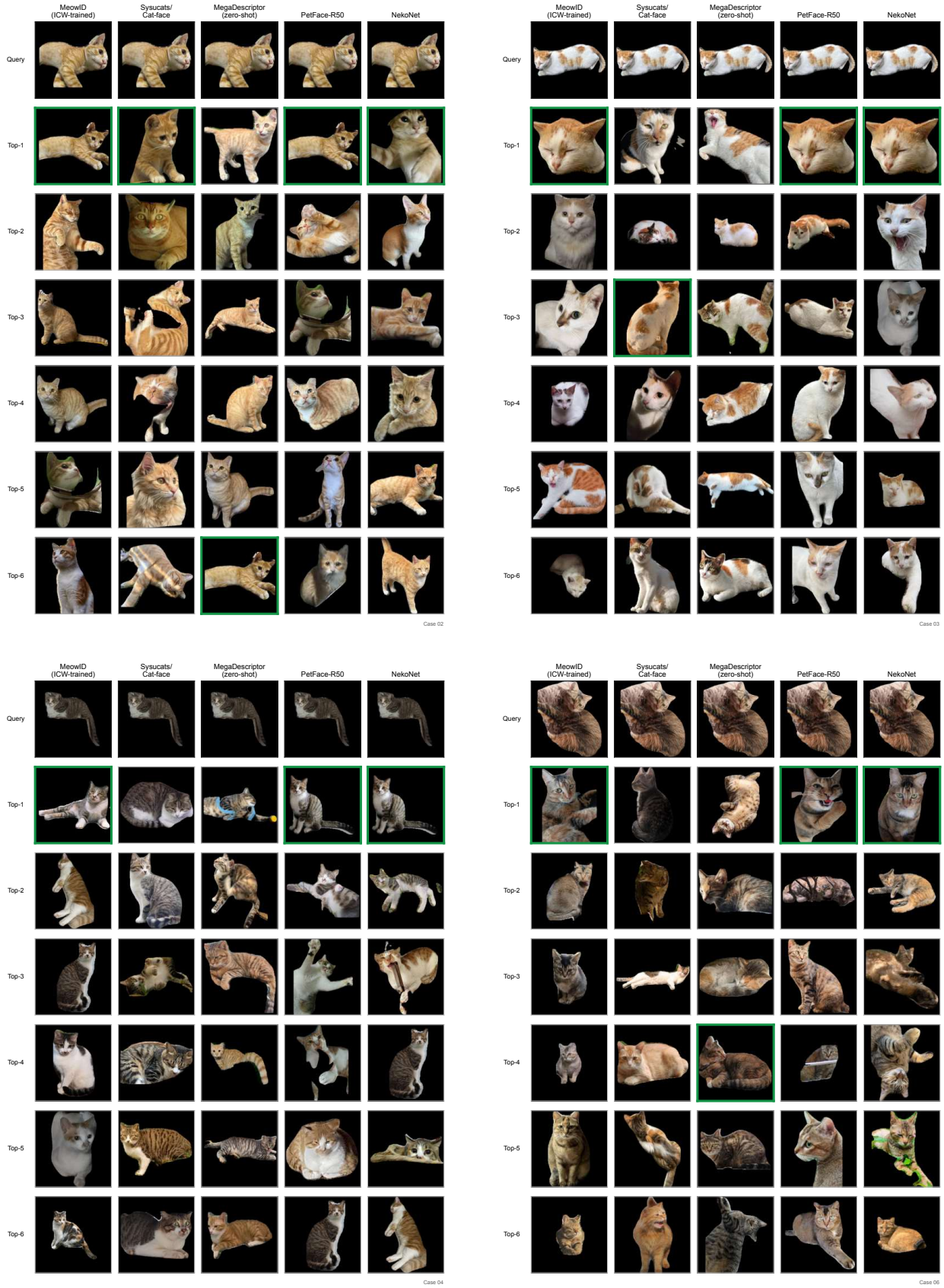


Figure 7 Qualitative comparison A.

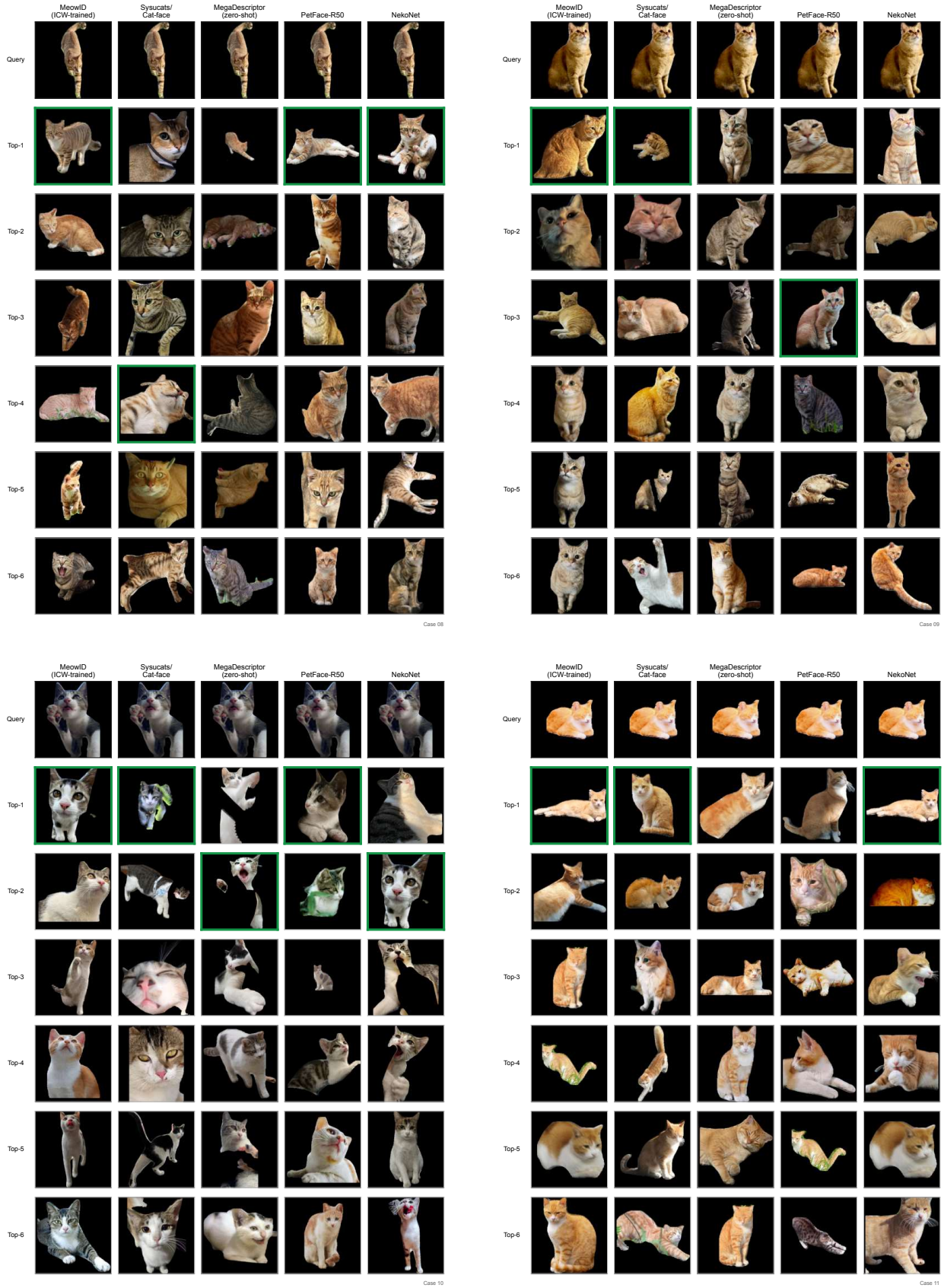


Figure 8 Qualitative comparison B.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- internaltop6directcomparison14cases1.jpg
- internaltop6directcomparison14cases3.jpg